Web Scraping and Ethical Considerations

Web scraping is a practice of automating the extraction of information that is available online which serves as an important means of producing data in several areas (Luscombe, Dick and Walby, 2021). As highlighted by Pagallo and Sciolla (2023), data scraping offers many beneficial uses, particularly for public-interest purposes. Individuals, journalists, and scholars employ scraping to efficiently gather and analyze large datasets, enhancing information accessibility. Pagallo and Sciolla (2023) argue that despite its beneficial uses, data scraping can also serve harmful purposes, such as spamming, website crashes, scams, privacy violations, and unfairly diverting website revenues.

Luscombe, Dick and Walby (2021) emphasize the complex ethical challenges that data scraping presents with no clear or consistent guidelines in place. Importantly, scraping can unintentionally harm websites—for example, by causing **denial of service (DoS)** attacks through excessive data requests. Something that we have to consider is to think and act ethically even if it means not getting data despite it being available or reducing the scope of our project to ensure access for others. In one case, researchers Boeing and Waddell (2017, cited in Luscombe et al., 2021) chose not to collect daily rental listing data from Craigslist, fearing that constant scraping would overload the site's servers, demonstrating ethical restraint in practice.

As outlined by Luscombe, Dick and Walby (2021) in their research, the best practices for web scraping includes respecting website limits and ethical considerations. One of the key guidelines is respecting the **robots.txt** file which outlines how bots should interact with a website. To reduce harm, scrapers should add delays (3-10 seconds), avoid peak hours, and consider the sensitivity of the data. Gregory (2018, , cited in Luscombe et al., 2021) stresses the importance of removing personal information, especially in sensitive fields like health. Ethical concerns go beyond formal approval and include situational ethics, where scrapers must weigh context-specific factors like potential harm and the public interest. As Tracy (2010, cited in Luscombe et al., 2021) argues, ethical research involves thoughtful decision-making, not just compliance. Ultimately, whether scraping is ethical depends on the data, purpose, and the researcher's values, and researchers must be prepared to justify their actions responsibly.

Web Scraping and Legal Considerations

Web scraping raises significant legal concerns, particularly as many websites now include terms and conditions that explicitly prohibit automated access (Din, 2015; Drivas, 2019; Scassa, 2019, all cited in Luscombe et al., 2021). For example, Canada's business registries service, although publicly accessible, requires users to agree not to use automated tools to copy data. Violating such terms may result in legal action, including cease and desist letters or lawsuits claiming unauthorized access or copyright infringement.

Luscombe et al. (2021) argue that website terms and conditions are often not designed with academic or public-interest research in mind, rather for business or legal protection. Therefore, researchers should not accept these restrictions uncritically. Instead, they advocate for "algorithmic thinking in the public interest"—a mindset that supports ethically justified scraping when it serves scholarly or civic purposes, even if legal boundaries are unclear. This approach

urges social scientists to question why certain data access restrictions exist, particularly when imposed by powerful governments or corporations, and to weigh potential legal risks against the societal value of conducting critical and investigative research (Haggerty, 2004; Nader, 1968; Galliher, 1979, all cited in Luscombe et al., 2021).

Key Takeaways for our Web Scraping Project

As a responsible global citizen, we should apply professional ethics, responsibilities, and norms of professional computing practice including awareness of regulation and ethical implications of acquisition, use, use, disclosure and eventual disposal of information. In conducting our web scraping capstone project, it is crucial to navigate both the ethical and legal dimensions of automated data collection with care and responsibility. Legally, while scraping publicly accessible data is not always clearly prohibited, many websites impose restrictions through their terms and conditions. Violating these terms can lead to consequences, even if legal actions against academic researchers remain rare. Ethically, scraping practices must avoid causing harm, such as overloading servers or collecting personal data, and should respect the broader digital ecosystem. Drawing from Luscombe et al. (2021), we recognise the importance of "algorithmic thinking in the public interest"—a principle that encourages researchers to critically assess data access restrictions and justify their scraping when it serves a public or scholarly purpose. Our approach to web scraping, therefore, should be guided by a commitment to responsible data use, transparency, and respect for both the technical limitations and the societal implications of our work.

How to employ ethical web scraping practices in our project?

Adhering to the terms and conditions of the website: We should always go through the terms and conditions of use of the website which we are using to scrape the data from and critically assess the legal and ethical considerations.

Scrape only public information in moderation: We should only scrape the publicly available information that does not require us to log in to the website or explicitly require us to accept terms and conditions.

Reading and respecting the robots.txt file: We need to go through the robots.txt file of the website and respect what is outlined for the bots behavior. We can integrate the behavior of reading and respecting the robots.txt file of a website in our web scraping script using Python. We must not crawl pages that are not allowed.

```
# Source: ScrapingAnt Blog - How to bypass Incapsula
# URL: https://scrapingant.com/blog/incapsula-bypass
# Accessed on: April 1, 2025
import requests
from urllib.robotparser import RobotFileParser
from urllib.parse import urlparse
```

```
def can_fetch(url, user_agent='MyBot'):
    rp = RobotFileParser()
    parsed_url = urlparse(url)
    robots_url = f'{parsed_url.scheme}://{parsed_url.netloc}/robots.txt'
    rp.set_url(robots_url)
    rp.read()
    return rp.can_fetch(user_agent, parsed_url.path)

# Example usage
url = 'https://example.com/some-page'
if can_fetch(url):
    print(f'Scraping {url} is allowed')
else:
    print(f'Scraping {url} is not allowed')
```

Seek Permissions: If ideal, request for a written consent from the owner of the website specifying what can be scraped.

Consider Copyright and Fair Use: Our scraping should not infringe on copyrights and make sure we make fair use of the website and content.

Respect the boundaries: Scrapers can significantly slow down the website which can impact human visitors or even crash the website. So, we need to go slow and scrape the website when it is not very active. Peak times should be avoided. We should respect rate limits, and add delays between requests.

Use a realistic and customized User-Agent: When sending HTTP requests or using Selenium-based automation, it's important to set a custom User-Agent string. This helps identify our script in a way that is ethical, transparent, and less likely to be blocked.
Example:-

```
options.add_argument("user-agent=Mozilla/5.0 (X11; Linux x86_64); Price
Tracker Bot; Contact: yourname@email.com")
```

Add Rate Limiting: We should add realistic delays of 3 to 10 seconds between requests instead of hammering servers with requests. If the website mentions the crawl-delay in the robot.txt file, we can take that into consideration while randomizing delays.
Example:-

```
import time
import random
time.sleep(random.uniform(3, 10)) //random delay of 3 to 10 seconds
```

Rotate IP Address and User-Agents: As noted by Finddatalab.com (2016), websites with anti-scraping protection can detect and block IP addresses that send too many repeated requests. The exact limit is unknown — only the website's system admin knows. To reduce the risk of being blocked:

- Rotate IP addresses so your scraper doesn't appear to come from a single source.
- Ensure IPs are from different ranges/networks, not just variations of the same.
- Combine IP rotation with user-agent rotation to simulate requests from different users and devices.

Use API if available: Use APIs to scrape data if the website provides an API.

References

Luscombe, A., Dick, K. and Walby, K. (2021) 'Algorithmic thinking in the public interest: navigating technical, legal, and ethical hurdles to web scraping in the social sciences,' *Quality & Quantity*, 56(3), pp. 1023–1044. https://doi.org/10.1007/s11135-021-01164-0.

Pagallo, U. and Sciolla, J.C. (2023) 'Anatomy of web data scraping: ethics, standards, and the troubles of the law,' European Journal of Privacy Law & Technologies, (2), pp. 1–19. https://doi.org/10.57230/ejplt232ps.

Finddatalab.com. (2016). The Ultimate Guide To Ethical Web Scraping and internet scraping. [online] Available at: https://finddatalab.com/ethicalscraping [Accessed 1 Apr. 2025].